DOCUMENT RESUME

ED 277 728 TM 870 004

AUTHOR Owston, Ronald D.
TITLE York Educational Software Evaluation Scales, Document Number 2.
INSTITUTION York Univ., North York (Ontario). Faculty of Education.
PUB DATE 1 May 85
NOTE 16p.
PUB TYPE Reports - Descriptive (141) -- Tests/Evaluation Instruments (160)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Computer Simulation; Course Content; *Courseware; Elementary Secondary Education; *Evaluation Methods; Evaluation Problems; Foreign Countries; *Instructional Material Evaluation; Postsecondary Education; *Rating Scales
IDENTIFIERS Canada; *Courseware Evaluation; *York Educational Software Evaluation Scales

ABSTRACT

According to a recent survey, the quality of educational software is the single most stated concern of computer educators in Canada. Only a small fraction of the educational software is evaluated. The evaluation approaches that have emerged to deal with this problem are subjective and unreliable, and they do not provide an overall impression of a given piece of software. The York Educational Software Evaluation Scales (YESES) presented in this document have been designed to overcome these problems. Drill and practice, tutorial, and problem solving software are rated on four key characteristics: pedagogical content, instructional presentation, technical adequacy, and documentation. A fifth scale, modelling, is used to rate simulation software. Each of these characteristics is rated on a four-point scale, with level 4 being indicative of a high rating on the characteristic, and level 1 being indicative of a low rating. The YESES are criterion based in that each level on the scale is referenced to a set of descriptors which give typical features of the software at that level. With these scales, the evaluation of software becomes the process of determining which set of descriptors best characterizes the software. Detailed directions for using the YESES are provided, including definitions of each of the four key characteristics of the sofware for each level of the scales. (JAZ)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED277728

YORK EDUCATIONAL SOFTWARE EVALUATION SCALES

by

Ronald D. Owston
Project Director

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

R. Owston

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy

Document Number 2

May 1, 1985

Faculty of Education
York University
North York, Ontario

TM 810 004

This document was produced and is distributed by York University, as a result of a Cooperative Project Agreement with IBM Canada Limited, whose support is acknowledged with thanks.

Limited reproduction for educational purposes may be undertaken provided appropriate acknowledgement is given.

INTRODUCTION

Estimates of the number of educational software titles currently on the market range from 8000 to 10,000, of which only a fraction have been evaluated. According to a recent survey (Scott, 1985), the quality of this software is the single, most-stated concern of computer educators in Canada.

This concern is shared by others as well. Alberta Education (1984) found that they were unable to recommend about nine out of ten software products they previewed for use in provincial schools. In the United States, Educational Products Information Exchange (EPIE) reported that only five percent of the educational software on the market could be rated as "exemplary" (MICROgram, 1984). While many knowledgeable educators found this figure unacceptably low, EPIE later confirmed it when they applied the new California State Department of Education "Guidelines for Educational Software for California Schools" "indicators of excellence" to a representative sample of software (MICROgram, 1984/85).

To address the problem of software quality, educators have taken two main approaches--software development and software evaluation. Software development can be seen in the exemplary software project of the Ontario Ministry of Education and the Minnesota Educational Computing Consortium.

The software evaluation approaches that have emerged can be organized along a continuum (Baker, 1983). Four points on this continuum can be identified:

- Organized networks with large numbers of evaluators using given sets of guidelines such as MicroSIFT and EPIE. Results of these evaluations are widely disseminated through directories, professional journals, and on-line databases.

- Subscription publications such as **Courseware Report Card** and **Software Reports** that do not have formal networks of evaluators, but instead rely both on in-house and out of house evaluators.

- The Minnesota Educational Computer Consortium (MECC), SOFTSWAP, and CONDUIT. The primary function of these organizations is software development, yet evaluation is an important ingredient in the development process. Neither MECC nor SOFTSWAP publish their evaluations; however, CONDUIT's evaluations are available by subscription.

- A category of discrete evaluation forms that individuals or groups may freely use. Typical of these forms are those of the National Council of Teachers of Mathematics and **Electronic Learning** magazine.

A number of problems can be identified in these software evaluation approaches, however. First, current approaches tend to be normative in nature. That is, evaluators are asked to rate on a scale various characteristics of software according to their strength of agreement with a statement about the software, or to give their written opinions on various aspects of the software. Since these evaluations are not based upon explicit criteria, evaluators tend to judge software relative to other similar software they have seen. A result of this method of evaluating software is that the ratings are subjective. This will not pose a problem if the evaluators are well-known authorities whose opinions are highly valued. A problem does occur in the interpretation of the evaluation results, however, if the evaluator is unknown to the reader.

Another related problem with current approaches to software evaluation is their inherent lack of reliability. Because the evaluations are subjective, consistency is unlikely from evaluator to evaluator using any given approach. Therefore any two evaluators would not necessarily rate a given piece of software the same, thus making it difficult for the reader of the evaluations to make a judgement about the software. Worse still is the case in which two different evaluators rate two similar pieces of software. The reader is then faced with an even more difficult task when interpreting the two evaluations.

Obtaining an overall impression of a piece of software is another problem with present evaluation approaches. These approaches typically require the evaluator to answer large numbers of questions about the software, yet provide little, if any, guidance in interpreting the discrete answers into a meaningful whole. The evaluation scales that do produce a global rating do so simply by taking weighted total score which cannot be interpreted very meaningfully.

The York Educational Software Evaluation Scales (YESES) presented in this document have been designed to overcome these weaknesses. Four important characteristics of drill and practice, tutorial, and problem solving software are rated on criterion-based scales with YESES. A fifth scale is used to rate simulation software. Each point on these scales is defined by a set of descriptors against which the software is judged. YESES is used by panels of evaluators who rate the software according to uniform guidelines. Further details on how YESES is used by the Faculty of Education and sample evaluation results can be obtained from Document Number 3.

Development of YESES would not have been possible without the valuable feedback of the Faculty of Education Project Planning Group--Andrew Effrat, Curt Dudley-Marling, Hugh McCully, Frank McIntyre, Angela Presutto, Mildred Shaw, and Laverne Smith--and the panels of teachers and consultants who have used the scales in various draft versions. Finally, much credit is due to the support and encouragement of members of the York/IBM Advisory Committee not already mentioned --Richard Blumenauer, Mike Brothers, Ken Fockler, William Found, Sheldon Levy, Bob Smithson, and Ian Taylor.

References

Alberta Education (1984). Curriculum courseware evaluations: December 1984. Edmonton, Alberta: Author.

Baker, N. B. (Ed.). (1983). Evaluation of educational software: a Guide to Guides. The Northeast Regional Exchange.

MICROgram (1984, October). Water Mill, N.Y.: Educational Products Information Exchange.

MICROgram (1984/85, December/January). Water Mill, N.Y.: Educational Products Information Exchange.

Scott, G. (1985). The PepperWood el-hi report on computers in Canadian education 1980-1986. West Hill, Ontario: PepperWood, Inc.

YORK EDUCATIONAL SOFTWARE EVALUATION SCALES

The York Educational Software Evaluation Scales are used to rate the educational value of microcomputer software. Drill and practice, tutorial, and problem solving software are rated on four key characteristics--pedagogical content, instructional presentation, technical adequacy, and documentation. Simulation software is rated on the same four characteristics as well as a fifth characteristic, modelling.

Each of these characteristics is rated on a four-point scale, with level 4 being indicative of a high rating on the characteristic, and level 1 being indicative of a low rating on the characteristic. Unlike other software evaluation schemes, the York scales are criterion based; that is, each level on the scale is referenced to a set of descriptors which give typical features of the software at that level. Thus, with these scales, the evaluation of software becomes the process of determining which set of descriptors best characterizes the software.

Directions:

1. Thoroughly familiarize yourself with the software by first emulating a very able student to test the limits of the software, and second, by emulating a weak student to find out how the software handles errors.

If during this step you find that the software contains racial, ethnic, or gender bias or stereotyping, discontinue the rating, skip to step 4, and indicate why you feel the software should not be rated. Do the same if you find that the software contains serious programming errors that prevent it from operating as described in the documentation.

2. Study the definitions of each of the four (or five) key characteristics of the software together with the descriptors for each level of the scales.

3. For each characteristic, select the level that best describes the software being rated. Undoubtedly you will find that none of the descriptors perfectly describes the software you are evaluating. Nevertheless, select the level that has the most features in common with, or is most similar in nature to, the software you have. Also, try to rate each characteristic independently, not allowing the rating on one characteristic to affect the rating on another.

4. After you rate the software on all four characteristics, write brief notes (maximum 200 words) on your overall opinion of the software. Try to make these notes as constructive as possible; for example, wherever appropriate, point out ways in which the software could be improved, how it could be integrated into the curriculum, or what the implications might be if it were used in the classroom.

CONTENT

Definition

Content refers to the knowledge and skills the software purports to teach--their organization, accuracy, and appropriateness. Content organization includes such aspects as the sequencing of the knowledge and skills within the lesson or lessons, the breadth or scope of the skills and knowledge, and the depth or intensity of instruction or practice given to a topic. Accuracy is concerned with truthfulness of the knowledge and skills that are presented. Appropriateness deals with the suitability of the content for the intended user which includes such factors as readability of the content, the match between the complexity of the content and the intended user's ability to master it, and the educational value of the content-- the time spent learning the content is justified because of its inherent value. The extent to which one, or all of these factors--organization, accuracy, and appropriateness--is weak is an indication of less than exemplary content.

LEVEL 4: Exemplary content

Level 4 content is superior in its organization, accuracy, and appropriateness. The content organization is such that the scope of the knowledge and skills is congruent with the user's ability to master them, the sequencing is logical and and follows good pedagogical practice (e.g. concrete presented before abstract), and the depth is sufficient to give the user adequate practice before proceeding to the next topic. The accuracy of level 4 content is extremely high. Furthermore, the content at this level is very readable, well-matched to the intended users's abilities, and has high educational value.

LEVEL 3: Desirable content

The organization, accuracy, and/or appropriateness of level 3 content is not quite as favourable as that of level 4 due to relatively minor weaknesses. The organization may be weak because the scope is not quite congruent with the user's ability to master it, the sequencing may be slightly illogical in several places or not quite in keeping with accepted pedagogical practice, or the depth may be either slightly more than necessary thus requiring the user to complete redundant exercises, or the depth may not be great enough so that the user does not receive sufficient practice before moving to the next topic. Problems with accuracy might consist of questionable (but not incorrect) facts or applications of concepts. Another possible class of difficulties with level 3 content is that there may be some vocabulary or sentence structures that may give some intended users difficulty, the knowledge or skills may be slightly too complex or too easy for the intended user, or some aspects of the content may be of slightly questionable educational value.

LEVEL 2: Minimally acceptable content

Level 2 content is clearly weak in either one, or a combination of, organization, accuracy, or appropriateness. The deficiency, however, is not serious enough to prevent the use of the software if no other better software is available and if the instructor is able to intervene to rectify the deficiency. Typical organizational problems found with level 2 software include the scope much greater than the user is able to deal with comfortably, the sequence poorly arranged or not consistent with good educational practice, or the depth considerably more or less than necessary. The kinds of accuracy problems encountered with level 2 content include incorrect minor facts or applications of concepts. The appropriateness problems found at this level include vocabulary and structure too difficult for most intended users, the knowledge and skills too difficult (or too easy), or the educational value of the content as a whole may be questionable.

LEVEL 1: Deficient content

Content at level 1 is sufficiently deficient so as to call into question the use of the software, regardless of the strengths of its other characteristics. Organizational problems may include weak, illogical sequencing, and scope and/or depth poorly matched with the user's ability. This level of content may also contain factual inaccuracies or incorrect applications of concepts. The content may not be very appropriate due to the reading level being considerably out of match with the user's ability, the knowledge and skills much too complex or simple, or the topics introduced by the software may be of very dubious educational value.

INSTRUCTION

Definition

Instruction is concerned with the manner in which the software takes advantage of the unique capabilities of the microcomputer in presenting the pedagogical content. These capabilities include the ability to be an interactive learning device, the ability to integrate text, graphics, and sound into the learning experience, the ability to give immediate feedback on the user's mastery of the content, and the ability to adapt the instruction to the user's needs. Instruction is also concerned with the psychological climate of the presentation--the sense of control of the lesson felt by the user, the tone of the instruction, and the supportiveness of the feedback. Instruction is not concerned with the quality of the screen displays nor with the sound quality.

LEVEL 4: Exemplary instructional presentation

Software at this level presents the content in a manner which takes maximum advantage of the unique capabilities of the microcomputer for which it was designed. That is, the content is presented in a highly interactive format; the software integrates text, graphics, and sound wherever appropriate; immediate, positive feedback is given to the user; and the software reasonably adapts to the user's needs by branching to an appropriate level of difficulty so that unnecessary routines do not have to be performed before proceeding to the next level of difficulty. Also, with level 4 software, the user has a sense of being able to control the software--to progress, to review, and to exit at will. The tone of the software is very supportive and non-punitive as evidenced by the language used in the instruction and the nature of the feedback given to the user's responses.

LEVEL 3: Desirable instructional presentation

Software at this level of instruction still presents the content so as to take advantage of the uniqueness of the microcomputer, however it does not do this as effectively as level 4 software. This may be due to such factors as: minor weaknesses in the interactive nature of the software; the integration of graphics, sound, and text could be improved upon slightly given the capabilities of the particular microcomputer; feedback may not be as immediate and constructive as desirable; or the branching may not be as adaptive as practicable. Other reasons why this software is not rated at level 4 include less user control over progress, ability to review, and ability to exit/reenter at will, and the tone of the software may not be as supportive and non-punitive as it might be.

LEVEL 2: Minimally acceptable instructional presentation

Level 2 software generally does not make effective use of the unique features of the microcomputer as an instructional device. While some aspects of the instruction may rate at level 3 or level 4, the overall quality of instruction is not as high as level 3 because of one or more distinct weaknesses in the instruction. Typical weaknesses might be: little or no attempt to integrate graphics, text, and sound; delayed or negative feedback to user responses; and little or no branching. Besides these weaknesses, the software may allow little control over the various aspects of its operation, and the tone may be either indifferent or slightly punitive.

LEVEL 1: Deficient instructional presentation

Software at this level, the lowest on the scale, exhibits no attempt-- or an unsuccessful attempt--at making use of the unique features of the microcomputer to present the content. Often this level of software requires the microcomputer to function as little more than a page turning device. There may be no integration of graphics, sound, and text (if, indeed, the first two features are used), the feedback may be non-existent or rewarding incorrect responses more than correct responses, and users may have to follow a lock step sequence to complete the lesson with this level of software. The software also may not allow the user to have any control over the operation of the software once the lesson has begun, and the tone may be punitive or non-supportive.

DOCUMENTATION

Definition:

Documentation refers to the supporting materials and instructions that accompany the software. This includes the printed materials provided as well as the supporting information available on the screen. One purpose of the documentation is to describe and explain how the software may be used pedagogically, which typically requires information on how the software may be integrated into the curriculum, the objectives of the software, the prerequisite skills necessary to successfully use the software, the age/grade/ability level recommended for the software, and suggestions and references for preparatory and follow-up activities. Another purpose of the documentation is to explain how to use the software in a technical sense, such as how to boot the disk, how to stop/start/reenter the program, and how to access various program components. From a pedagogical point of view, both of these purposes must be served by the documentation.

LEVEL 4: Exemplary documentation

Software at this level has clearly written, concise documentation that fully explains how the software may be used pedagogically and technically. Level 4 documentation contains thorough information on how the software may be integrated into the teaching/learning process by providing complete details on aspects such as the prerequisite skills and abilities required, follow-up activities, and possibly worksheets, if appropriate, and bibliographic references. The documentation must also explain all aspects of the operation of the software including--but not limited to--how to boot the disk, stop/start/reenter the program, use peripheral devices, and branch to components of the software so that the user does not have to spend needless time on trial and error. The main criterion for rating the documentation level 4 is if the software can be readily used both pedagogically and technically after a carefully reading of all the documentation.

LEVEL 3: Desirable documentation

Level 3 software documentation, like level 4, describes and explains how the software may be used both pedagogically and technically. Unlike level 4 documentation, however, level 3 documentation is not as high in quality in its suggestions on how the software may be integrated into the teaching/learning process. This may be due to minor omissions in some pedagogical aspects, or due to a lower overall standard in the pedagogical information. The technical documentation would normally be expected to be the same quality as level 4.

LEVEL 2: Minimally acceptable documentation

This level of documentation contains a minimal amount of usable pedagogical information, while the technical information may have some minor errors or omissions. Typically this level of documentation requires considerable effort on the part of the teacher in order to integrate the software into the curriculum. Users may find that technical problems, depending upon their computer background, have to be resolved through trial and error.

LEVEL 1: Deficient documentation

Documentation at this level is clearly inadequate to support the use of the software either technically, pedagogically, or both. Pedagogical documentation is non-existent or clearly unusable at this level; technical documentation may be confusing, imprecise, and incomplete.

TECHNICAL ADEQUACY

Definition

Technical adequacy refers to the overall quality of the design of the software with respect to user inputs, the software's outputs, and systems errors. More specifically, adequacy of input is concerned with the ease and simplicity of entering data into the computer. The technical adequacy of outputs refers to the aesthetic quality of the sound, graphics, and colour given the limitations of the machine for which the software was designed, as well as the clarity of the screen layout. System errors means the ability of the software to continue to operate and not "lock up" regardless of the kinds of inputs the user enters.

LEVEL 4: Exemplary technical adequacy

The technical adequacy of level 4 software is extremely high. This level of software gives clear, precise cues and prompts so that the user knows when, and in what form, the input is required. Level 4 software avoids the use of characters with special meaning, requires a minimal amount of typing (except if the objective of the lesson is typing proficiency), does not distinguish between upper and lower case characters unless there is a pedagogical rationale to do so, accepts partial or abbreviated answers wherever appropriate (i.e. Y for Yes, N for NO), allows responses to be corrected before they are accepted by the machine, and restricts the input to the same location on the screen. Also, the colour, sound, and graphics of level 4 software are aesthetically pleasing, and the screen layout is uncluttered and consistent from screen to screen. Finally, level 4 software is of such technical quality that no system errors whatsoever are evident when running the software.

LEVEL 3: Desirable technical adequacy

This level of software is not as technically adequate as level 4 software due to minor flaws in its design. The flaws, however, may be regarded as slight inconveniences, not serious enough to detract from efficient learning. Initially, the user may be uncertain with some aspects of the input to level 3 software, such as its form and location, but after some trial and error or reading of the accompanying documentation, the uncertainty can be cleared up. For example, the software may be occasionally inconsistent in accepting abbreviated responses, or there may be inconsistencies from time to time in the location of input. Other weaknesses of level 3 software may be that the colour, sound, and graphics are not as aesthetically pleasing as level 4 software, the screen layout may not be quite as consistent from screen to screen as it might be, or the screen may be slightly cluttered. Like level 4, level 3 software should not contain any system errors.

LEVEL 2: Minimally acceptable technical adequacy

Level 2 software has distinct weaknesses that are, at the least, constant annoyances to the user, and at most, detract from efficient learning. Often this level of software is inconsistent in the form and location of the input that is required. Unlike level 3 software, however, these inconsistencies are not always predictable by the user, thus requiring trial and error to determine the form and/or location of the input. Examples of these inconsistencies include accepting abbreviations at one point during the lesson but not another, distinguishing between upper and lower case letters sometimes but not others, and requiring user responses to be entered at different screen locations from one frame to the next. The colour, graphics and sound on level 2 software may be less than desirable aesthetically, or the screen layout may tend to be cluttered. Although there may be no actual programming errors with level 2 software, the program operation could be improved.

LEVEL 1: Deficient technical adequacy

Level 1 software typically has technical flaws that hinder efficient learning regardless of the content and instructional presentation. One reason for this may be because the form and location of the input varies from frame to frame even more frequently and less predictably than level 2 software. Poor colour, graphics, and sound, or cluttered and confusing screen layout may be other reasons why software may be classified as level 1. Further reasons may be that the software has actual programming errors that are detectable during normal operation.

MODELLING

Definition

Modelling refers to the adequacy of the model used in simulation software to simulate a real life situation. At least three factors must be considered when judging the overall adequacy of the simulation model. First, the complexity of the model relative to the intended user must be examined. Most real life situations are too complex to be modelled with even a large number of variables; therefore, a balance must be struck between the realism of the model and the ability of the intended user to deal with all of the variables in the model. Second, all of the variables in the model must be relevant to the simulation. For example the relevance of the lunar cycle as a variable in a stock market simulation is highly questionable. And third, the variables in a simulation model must interact and produce results similar to what would be expected in a reality, otherwise the user might develop a misleading view of reality. Thus to determine the adequacy of a simulation model one must examine its complexity relative to the intended user, its relevance, and its results.

LEVEL 4: Exemplary modelling

Software at this level provides a highly realistic portrayal of a real life situation without being too complex nor too simple for the intended user. The variables included in the model are the most relevant ones to use for the given simulation. Furthermore, all of the variables in the model interact and produce results approximately the same as they would do in real life.

LEVEL 3: Desirable modelling

Level 3 software has a less adequate, yet still a very usable simulation model. Possible weaknesses of the model include one or more of the following: The number of variables in the model may be slightly too great so as to make the model too complex for the intended user, or on the other hand, there may be too few variables which may tend to slightly trivialize the simulation for the intended user. One or two of the variables may be slightly irrelevant to the model, and therefore, they could be either eliminated or replaced by a more relevant variable or variables. Another weakness may be that the variables interact or produce results that are sufficiently different from reality to be slightly misleading, but not enough to warrant rating the software level 2.

LEVEL 2: Minimally acceptable modelling

The simulation model in level 2 software has some significant weaknesses, although the software is still usable in certain contexts. An example of one of these contexts would be if the instructor thoroughly explained the weaknesses of the software and supplemented it with additional materials or experiences. One typical weakness of level 2 software is that the model contains too few or too many variables, thus making the software either too difficult or too simple for the intended user. Another problem with the software may be that there are irrelevant or inappropriate variables that significantly decrease the quality of the simulation. A further difficulty with the model may be that the variables interact or produce results that differ substantially from the real situation that is being simulated, to the point of misleading the user and causing confusion.

LEVEL 1: Deficient modelling

Software rated level 1 on modelling is generally unusable regardless of its strengths in other areas. This is because the number of variables used in the model is far too few or too great to be appropriate for the intended user. Another reason could be that the variables included in the model may be totally inappropriate to provide a realistic simulation of a true life situation. A final reason why the software may be rated level 1 is that the variables interact or produce results that are totally unrealistic or misleading.